Valuing American Options with Implementation

by Kathrine Salamonsen

08.06.2017

Master Thesis in Financial Economics

Supervisor: Jochen Jungeilges

Abstract

This master thesis aims to value American put options by using different numerical methods. Three valuation methods for valuing an American put option will be presented and analyzed; the binomial method, the implicit finite difference method and the least squares Monte Carlo approach (LSM). Due to the opportunity of early exercise of American option contracts, our goal is to find the optimal exercise strategy which maximizes the payoff by using numerical methods. We provide examples of how to implement each algorithm in different types of software. A comparison of the methods are given at the end.

University of Agder (2017)
Faculty of (School of Business and Law at UiA)
Keywords: American Options, Least Square Monte Carlo, Binomial Method, Implicit Finite Difference Method
JEL-classification: C63, C61, G17, G12

Contents

1 Introduction **5**

2 Literature Review **7**

3 Valuation Methodology **8**

3.1 Binomial Method - Framework to Value an American Put Option . . . 9

3.2 Implicit Finite Difference Method 15

3.3 Least-Square Monte Carlo Method 22

3.3.1 Valuation Framework for Least Squares Monte Carlo 22

3.3.2 The LSM Algorithm 24

4 Implementation of Valuing American Options **26**

4.1 Implementation of the Binomial Method in Excel 26

4.2 Implementation of the Implicit Finite Difference Method in Excel 28

4.3 Implementation of Least Square Monte Carlo Method in STATA 32

4.4 Comparison of the Valuation Methods in Matlab 40

5 Conclusion **43**

6 References **44**

7 Appendix **46**

7.1 STATA: Generating a Stock Price Path to Use in the LSM 46

7.2 STATA: Program to Value an American Put Option in the Least Squares Monte Carlo Method 47

7.3 MATLAB: Program to value an American Put Option in the Binomial Method (Cox, Ross and Rubinstein) 49

7.4 MATLAB: Program to Value an American Put Option in the Finite Difference Method 51

7.5 MATLAB: Program to value an American put Option in the Least Squares Monte Carlo Method 53

7.6 MATLAB: Program to Compare the Valuation Methods 55

List of Figures

1 Simulated Stock Price Path for S1 5
2 A Three Period Binomial Tree of Stock Prices 10
3 A Three Period Option Payoff Tree 14
4 Structure of the Grid in the Finite Difference Approximation 17
5 Graphical Representation of Equation (22) 19
6 Lattice Tree of Possible Stock Prices and Option Values 27
7 Parameter Values 28
8 Stock Price Grid 29
9 Values Along the Boundary of the Grid 29
10 Coefficients $a(j)$, $b(j)$ and $c(j)$ 30
11 Tridiagonal Matrix A 30
12 Adjustment Vector d_i 31
13 Option Prices for the American Put Option 31
14 Stock Price Paths in STATA 32
15 STATA: Value of the European Put Option 33
16 STATA: Regression at Time 2 36
17 STATA: Regression at Time 1 38
18 STATA: Value of the American Put Option 39
19 Computational Time 42
20 3D Surface of the Valuation Methods 43

List of Tables

1 Parameters Used in the Binomial Method 26
2 Stock Price Paths 32
3 Cash-Flow Matrix at Time 3 33
4 Regression at Time 2 34
5 Optimal Early Exercise Decision at Time 2 34
6 Cash-Flow Matrix at Time 2 35
7 Regression at Time 1 36
8 Optimal Early Exercise Decision at Time 1 37
9 Stopping Rule 38
10 Option Cash Flow Matrix 39
11 Comparison of Results 40
12 Simulation of an American Put Option 41

1 Introduction

An American put option, is a contract that gives the holder the right, but not the obligation, to sell a specified asset (stock, bond, currency etc.) for a specified price at or until a specified time in the future. Due to the early exercise feature, the question for an optionholder is: when is it optimal to exercise the option? We could wait until the expiry date, but is this the optimal value of the option? Figure (1) gives a demonstration of the problem. In the figure, we see a simulated price path for a stock $S1$. The stock price series is starting at time $t = 0$ with a value of $S_0 = 1.0$. This time series plot shows how the stock price is moving over a time period of 100 observations. The risk-free rate is 5%, with a volatility of 32%. The red line is representing the strike price $K = 1.05$. Above the red line is the out-of-the money region, in other words, the option is worthless and the investor would let the option expire. He could also hold on to the option to see if the stock price decreases before expiry of the option. Below the red line, the option is in-the-money. Here, the investor has many opportunities to exercise the option. As the stock price decreases the option becomes more valuable. The payoff function for an American put option at maturity is given by $V_0^{put} = max(K - S, 0)$. Which exercise date is the optimal one? When is the payoff maximized?

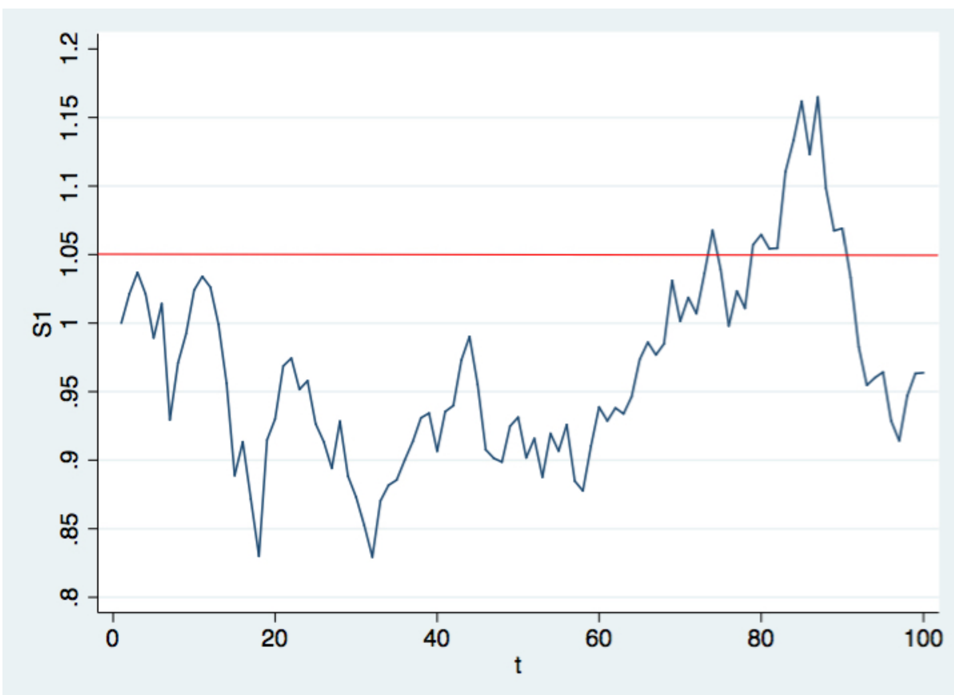

Figure 1: Simulated Stock Price Path for S1

This thesis aims to find the optimal stopping time for an American put option where the payoff is maximized by using different numerical valuation methods. We value an American option by using three different approaches. We present the binomial method by Cox, Ross and Rubinstein (1979), the finite difference method by Schwartz (1977) and the least squares Monte Carlo approach by Longstaff and Schwartz (2001). Each valuation method will be used in numerical examples and we shown how to implement them in various type of softwares such as Excel, STATA and Matlab. In the numerical analysis we shows how we find the optimal value of the American put and when the optionholder should exercise the option. Thereafter we compare prices of American put options obtained by the the different methods, and show that the early exercise value of the American put option is larger than the value at maturity for an European put option. At the end, we compare the methods in terms of computational time of how closely they value the option.

The structure of this thesis is as follows. In Section 2, literature is reviewed. Section 3 presents the valuation methodology for the three valuation frameworks. Examples of how to implement each method is given in section 4. Last section 5, concludes.

2 Literature Review

Pricing of American style options are usually attained by numerical methods, since there is no analytical solution available. A major breakthrough in option pricing was done by Black and Scholes in 1973, when they introduced the famous Black-Scholes equation to value options contracts. Their approach was mainly for European options which is solved analytically. A few years later, in 1977, Schwartz were the first to introduce the finite difference method, where the partial differential equation is transformed by approximation of the partial derivatives to a difference equation. Shortly after, in 1979, Cox, Ross and Rubinstein introduced the binomial tree, a method that constructs a stock price tree of upward and downward movements of the stock price and discounting the expected payoff at each node of the tree. Recently, in 2001, Longstaff and Schwartz introduced a new approach to value American options by using simulation. Their approach is called least squares Monte Carlo (LSM), where the conditional expectation function is estimated to find the expected value of continuing to hold the option.

On the topic of valuing American options, there is a large contribution in the literature. Since there are many different varieties of option contracts with different features of exercise policy, there has been considerable research on the topic. This master thesis differs in how it provides implementation to use the valuation methods in different softwares. Of literature, Lin & Liang (2007) price a perpetual Bermudan option and a perpetual American option by using the binomial method. They obtain a closed form solution and the optimal boundary condition for the options, and present a numerical experiment based on the pricing formulas they found. Stentoft (2004) gives a detailed analysis of the least squares method by Longstaff and Schwartz (2001). He analyzes how the LSM approach goes by increasing the number of stochastic factors. He concludes that for higher dimensional problems the LSM method should be preferred to the binomial method. Another article by Chen, Huang and Lyuu (2015) parallelize the LSM by space decomposition, where they analyze the accuracy and efficiency the parallelized LSM. They find that the option price obtained by the parallelized LSM are close to the values obtained by the sequential LSM and binomial tree, and by using parallel LSM the pricing of option is effectively speeded up and still is accurate.

3 Valuation Methodology

A popular topic in finance are derivative contracts. One type of a derivative, is the option contract. Options are actively traded in the financial market. An option is a contract to buy or sell a specific financial product. Options come in different versions, they can be simple or very complex. Two types of common option contracts are the European and the American. Kijima (2013) states that American options are at least as valuable as the European option because the exercise decision for an American option can be postponed until maturity. Also the possibility of exercising the American option at an earlier time, make American options more valuable than the European option. American options are more widely traded compared to European options. In 1977, Chicago Board Options Exchange added put options on their exchange board. An American *put* option gives its buyer the right but not the obligation to *sell* an asset for a specified price at or until a specified time in the future. Due to the opportunity of early exercise, the objective is to determine the optimal exercise strategy that maximizes the payoff of an American put option. For some options, such as American options, numerical methods are used in determining the value of the option.

Today, there exists a variety of different valuation methods accounting for the early exercise feature of American options. The most well-known methods used to value American options is the binomial tree, the finite difference method and the least squares Monte Carlo approach. In the binomial method, expected payoff is discounted recursively and compared with the value of immediate exercise. As for the implicit finite difference method, we can compute the option price by approximating the partial differential equation by a difference equation and solving the difference equation numerically. For the relatively new approach called least squares Monte Carlo, a conditional expectation function is estimated by least square regression, giving an estimated conditional expectation of the continuing value to hold on to the option. By using these valuation methods we can find an approximate value of the American option.

The question for a holder of an American put option is to decide *when* or *if* he should exercise the option. If the option is out-of-the-money at time t, he should not exercise the option. However, if the option is in-the-money it may be beneficial to exercise the option, or even wait longer because the payoff might be larger at a later time.

This section presents three different frameworks to value an American put option. Each valuation framework consider an American put option on a stock that pays no dividends. For simplicity of presenting the methodology and implementation of the valuation methods, we consider a short timeframe of $n = 3$.

3.1 Binomial Method - Framework to Value an American Put Option

This text[1] is based on Higham (2014). An elegant and easy way to value American options is the binomial method. The objective is to present a framework to value an American option by constructing a tree of stock prices and option prices. The framework will price the option and determine the optimal exercise strategy at every time step of the option.

We let $\delta t = T/n$, represent the timeframe in this model, where T is the expiry date of the option and n is the number of steps. Stock prices will be considered at times $t_i = i\delta t$ for $0 \leq i \leq n$. At time $t_0 = 0$, the initial price of the stock is S_0, which is a known number. At the next period this stock price will either go up by a factor u or down by factor d. This gives the prices of the stock in the next period at time $t_1 = \delta t$, that we denote as $S_0 u$ for an upward movement in the stock price and $S_0 d$ for a downward movement in the stock price. At time $t_2 = 2\delta t$ the stock prices will be $S_0 u^2$, $S_0 ud$ and $S_0 d^2$. In the last period $t_3 = 3\delta t$, the stock price will be $S_0 u^3, S_0 u^2 d, S_0 d^2 u$ and $S_0 d^3$. A demonstration of the movement of the stock price over $0 \leq n \leq 3$ time increments is given in Figure (2).

[1] Part of this text is based on Valeriy Zakamulin lecture notes from course BE-419 at University of Agder, Lecture 15: Pricing and Exercising of American Options in the Binomial Model.

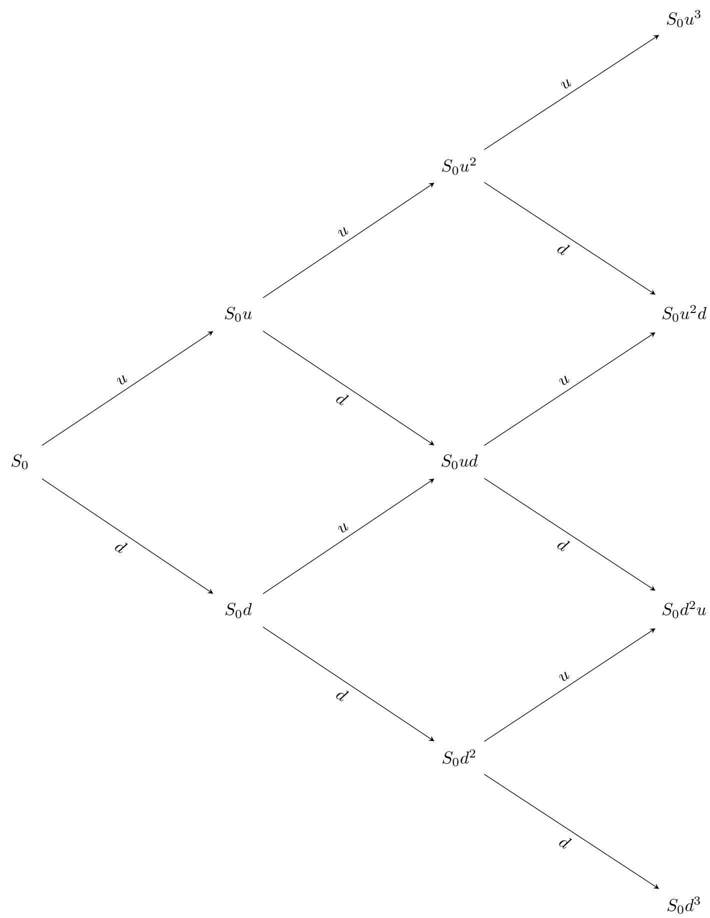

Figure 2: A Three Period Binomial Tree of Stock Prices

To obtain an expression for the up and down factor u and d in the binomial method we define a Bernoulli random variable R_i, with $\mathbb{E}(R_i) = p$ and $var(R_i) = p(1-p)$. If the stock price goes up, then $R_i = 1$ with a probability of p, and if the stock price goes down $R_i = 0$ with a probability of $(1-p)$. In the case of n time increments, the stock has

$\sum_{i=1}^{n} R_i$ upward movements and $n - \sum_{i=1}^{n} R_i$ downward movements. At time $t = n\delta t$, the stock price $S(n\delta t)$ is given by

$$S(n\delta t) = S_0 u^{\sum_{i=1}^{n} R_i} d^{n-\sum_{i=1}^{n} R_i}.$$

We re-arrange the equation by moving S_0 to the left, and fixing the expression on the right side

$$\frac{S(n\delta t)}{S_0} = d^n \left(\frac{u}{d}\right)^{\sum_{i=1}^{n} R_i}.$$

Then we take logs on both sides of the equation

$$\log\left(\frac{S(n\delta t)}{S_0}\right) = n \log d + \log\left(\frac{u}{d}\right) \sum_{i=1}^{n} R_i.$$

The Central Limit Theorem says that for large n, the sum $\sum_{i=1}^{n} R_i$ behaves like a normal random variable. Consequently, $\log\left(\frac{S(n\delta t)}{S_0}\right)$ will be close to normal for large n. We require the mean of $\log\left(\frac{S(n\delta t)}{S_0}\right)$ to be $(\mu - \frac{1}{2}\sigma^2)n\delta t$ and its variance to be $\sigma^2 n\delta t$ in order to match the continuous stock price model. If we impose the risk-neutrality assumption that $\mu = r$, we get two conditions:

$$p \log u + (1-p) \log d = (r - \frac{1}{2}\sigma^2)\delta t. \tag{1}$$

$$\log(\frac{u}{d}) = \sigma \sqrt{\frac{\delta t}{p(1-p)}}. \tag{2}$$

Equation (1) and (2) contains three unknown variables u, d and p. To find one possible solution we may set $p = \frac{1}{2}$, which gives

$$\frac{1}{2} \log u + \frac{1}{2} \log d = (r - \frac{1}{2}\sigma^2)\delta t. \tag{3}$$

$$\log(u) - \log(d) = \sigma 2\sqrt{\delta t}. \tag{4}$$

Multiply equation (3) by 2 and add equation (4), we get

$$2\log(u) = 2\left\{(r - \frac{1}{2}\sigma^2)\delta t + \sigma\sqrt{\delta t}\right\}$$

$$\Rightarrow \log(u) = (r - \frac{1}{2}\sigma^2)\delta t + \sigma\sqrt{\delta t}$$

By solving for u, we obtain the expression for u, an upward movement in the stock price

$$u = e^{(r-\frac{1}{2}\sigma^2)\delta t+\sigma\sqrt{\delta t}}. \tag{5}$$

For d, we multiply equation (3) by 2, but subtract equation (4), this gives

$$2\log(d) = 2\left\{(r - \frac{1}{2}\sigma^2)\delta t - \sigma\sqrt{\delta t}\right\}$$

$$\Rightarrow \log(d) = (r - \frac{1}{2}\sigma^2)\delta t - \sigma\sqrt{\delta t}$$

Solving the equation for d, we obtain the expression for d, the downward movement in the stock price

$$d = e^{(r-\frac{1}{2}\sigma^2)\delta t-\sigma\sqrt{\delta t}}. \tag{6}$$

It follows that $d < e^{r\delta t} < u$, otherwise there will be arbitrage opportunities that one can gain risk-less profit.

Cox, Ross and Rubinstein (1979)[2] have another solution of finding u and d. An assumption about the behavior of the underlying stock's stochastic process has to be done. We assume that the stochastic process is continuous as $n \to \infty$. The parameters must be chosen in a way to determine the right values of the expected return and variance of the stock at the end of each time interval, Δt. The one period expected return of the stock is equal to the risk-free rate $r\Delta t$ given the assumption about risk-neutrality, and the expected future price of the stock is $Se^{r\Delta t}$.

$$Se^{r\Delta t} = pSu + (1-p)Sd$$

and

$$e^{r\Delta t} = pu + (1-p)d. \tag{7}$$

[2]This paragraph is based on Alberto Barola (2013): "Monte Carlo Methods for American Option Pricing"

A one period variance of $\sigma^2 \Delta t$ is assumed in the stochastic process and formulated as

$$pu^2 + (1-p)d^2 - [pu + (1-p)d]^2 = \sigma^2 \Delta t. \tag{8}$$

Inserting p from equation (7) into (8) we get

$$e^{r\Delta t}(u+d) - ud - e^{2r\Delta t} = \sigma^2 \Delta t. \tag{9}$$

A third condition was introduced to derive the equations for u, d and p,

$$u = \frac{1}{d}. \tag{10}$$

Condition (7), (9) and (10) can be solved for each of the three unknown variables and gives

$$p = \frac{e^{r\Delta t} - d}{u - d}, \qquad u = e^{\sigma\sqrt{\Delta t}}, \qquad d = e^{-\sigma\sqrt{\Delta t}}.$$

Our task is to find the value of the option today at time zero, V_0. As the binomial method is recursive, we work backwards through the tree. An American put option can be exercised at any time step n but only exercised one time. At time 3 the option is represented by its payoff which has four possible values. The price of the option has three possible values. At time 2 the writer of the option can choose between exercising the option immediately or wait until time 3. Staring at the end of the tree and working backwards, the payoff functions at expiry $n = 3$ for an American put option are

$$Vu^3 = max(K - u^3 S, 0)$$

$$Vu^2 d = max(K - u^2 dS, 0)$$

$$Vd^2 u = max(K - d^2 uS, 0)$$

$$Vd^3 = max(K - d^3 S, 0)$$

Figure (3) represents the option payoff tree for three periods.

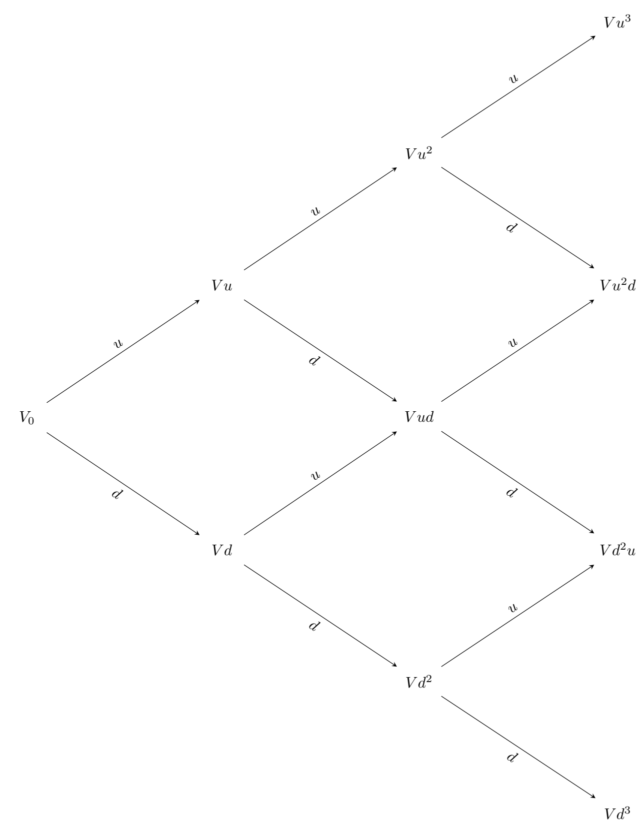

Figure 3: A Three Period Option Payoff Tree

Once the payoff at the final expiry date for each node is found, the next task is to compare the value of keeping the option V^{keep} with the value of exercising the option $V^{exercise}$ at each node in the tree.

$$V = max(V^{keep}, V^{exercise}) = max(e^{-r\delta t}(pVu + (1-p)Vd), K - S).$$

The optionholder chooses to exercise the option when the value of exercising is larger than the value of keeping the option, $V^{exercise} > V^{keep}$.

3.2 Implicit Finite Difference Method

This text is based on Kyng, Purcal and Zhang (2006) and Černý (2009). Another numerical method to value American options is the finite difference method. The finite difference method comes in different varieties. Examples are explicit, implicit and Crank-Nicolson. Since the explicit finite difference method can be unstable if we don't choose the value of discretization parameters carefully, we decided to use the implicit finite difference method as it is always stable. Stability of the implicit finite difference method will be discussed later in the text.

The objective by using the implicit finite difference method is that we approximate the partial differential equation (PDE) by using difference equations to solve the difference equation numerically. By working recursively, we find the option price. For American style option there are no closed form analytical solutions, hence, we have to solve the PDE numerically. First, construction of the stock price grid accounting for the boundary conditions for an American put option will be given. The next step is to solve the PDE by finding expressions for the derivatives and partial derivatives in the PDE equation. The last step is to solve a set of linear simultaneous equations by matrix algebra.

Consider a stock price process following the risk-neutral stochastic differential equation (SDE)

$$dS = rSdt + \sigma SdZ,$$

where r and σ are constants, Z is a standard Brownian motion. The well-known partial differential equation (PDE) by Black and Scholes (1973) is given by

$$\frac{\partial F}{\partial t} + (r-y)S\frac{\partial F}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 F}{\partial S^2} - rF = 0, \tag{11}$$

For an American put, the boundary conditions are

$$F(S,T) = max(K-S,0), \tag{12}$$

$$F(0,t) = Ke^{-r(T-t)} - Se^{-y(T-t)} \tag{13}$$

and

$$\lim_{S\to\infty} F(S,t) = 0, \tag{14}$$

then the solution to the PDE equation (11) is given by

$$F(S,t) = Ke^{-r\tau}N(-d_2) - Se^{-y\tau}N(-d_1), \tag{15}$$

F is the option price defined on the domain $D = \{(S,t) : S \geq 0,\, 0 \leq t \leq T\}$. t stands for time, S is stock price, K is strike price, y is dividend yield, r is the risk-free rate, the volatility of the stock is represented by σ and T is the maturity date. $N(\cdot)$ is the cumulative distribution function of the standard normal distribution. $\tau = T - t$ represents the time to expiry, when t goes from 0 to T.

To obtain the finite difference approximated form of the required partial derivatives, we need to define the increments ΔS and ΔT. Assume N equally spaced time intervals until the expiry of the option at T. ΔT will then represent the length of each interval, that is $\Delta T = T/N$. The boundary conditions for the American put option need to be adjusted before we can specify the stock price increment ΔS. Boundary condition (13) applies for $S \leq S_{min}$, and $S \geq S_{max}$ applies for boundary condition (14).

When the stock price S is extremely high, the American put option is deeply out of the money. Assume there exists a stock price S_{max} such that $S \geq S_{max,}$ the put option is deeply out of the money with a value approximately zero

$$S \geq S_{max} \Rightarrow F(S,t) = 0. \tag{16}$$

Equation (16) is another way of representing equation (14), this is the boundary condition at the bottom of the grid to be constructed. Usually the boundary is set to be $S_{max} = 2 \cdot S_0$. Conversely, when the stock price is extremely low, the American put option is deeply in the money. Assume there exists a low stock price $S \leq S_{min}$, making the option deeply in the money and certain to be exercised at expiration. Hence, the option can be regarded approximately as a forward contract

$$S \leq S_{min} \Rightarrow F(S,t) = Ke^{-r(T-t)} - Se^{-y(T-t)}.$$

Usually $S_{min} = 0$, so that

$$S = S_{min} \Rightarrow F(S,t) = Ke^{-r(T-t)}. \tag{17}$$

Equation (17) is the boundary condition for the top of the grid. Using equation (16) and (17) leads us to define the stock price increment as $\Delta S = (S_{max} - S_{min})/M$. Now we are able to create a grid of the change in stock prices and times. Time is indexed by i and j is indexes the stock price level. The range of values for i is $i = 0, 1, 2, ..., N$, so that for i there are $N + 1$ different values of time. For j, the range of values is $j = 0, 1, 2, ..., M$, with $M + 1$

different values for the stock price level. A discretized version of function F, can be defined as $f(i,j) = F(j \times \Delta S, i \times \Delta T)$, so that there is $(M+1) \times (N+1)$ different values of the function $f(i,j)$. By using finite difference approximation to the partial derivatives in PDE, the function $F(S,T)$ differential equation becomes a difference equation for $f(i,j)$, and will be shown soon. Figure (4) [3] is a representation of the grid of stock price levels and time step increments.

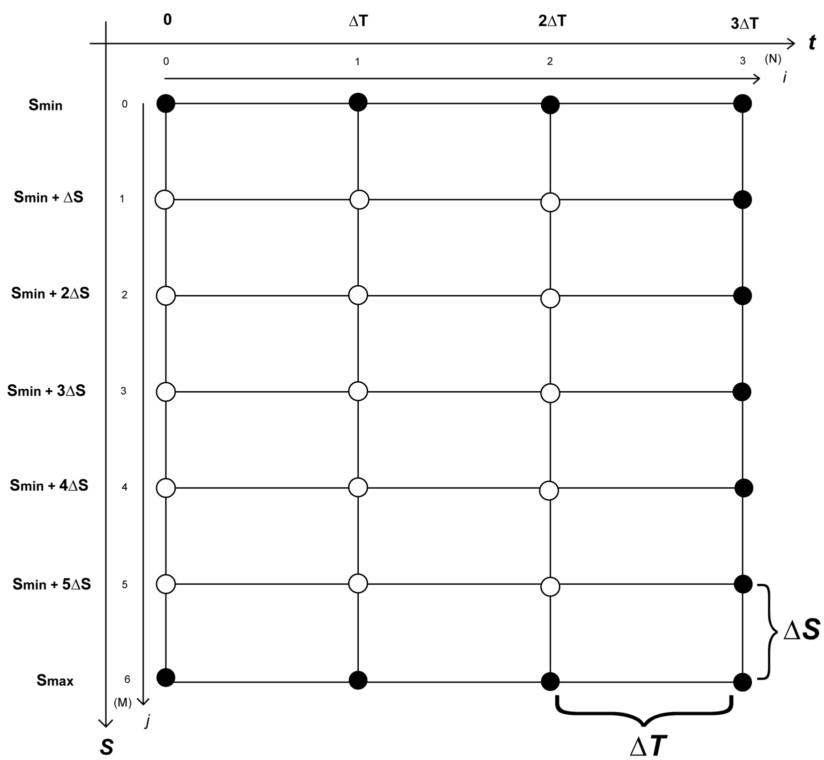

Figure 4: Structure of the Grid in the Finite Difference Approximation

On the horizontal axis, we have time t, where each step increases from left to right by ΔT. Time steps indexed by i, runs from 0 to the largest value of N. The vertical axis shows the stock price S, starting from the top at S_{min} and the stock price level increases as we reach the bottom, S_{max}. ΔS is showing each step in the stock price level. Stock price steps are indexed by j, and runs from 0 to the largest value of M. The white circles represents the values we want to find. On the upper, left and bottom of the grid, we have black circles, here the values are known from the boundary conditions (17), (12) and (16) .

Now we need to find the approximate difference equations of the derivatives. By using Taylor's expansion, we can obtain an approximation for the partial derivatives in PDE (11)

[3] Illustration of the stock price grid is made by Kathrine Salamonsen in Adobe Illustrator CS3 and has inspiration from Kyng, Purcal and Zhang (2016) Figure 1, page 6.

equation. Recall that the discretized version of F is $f(i,j)$, then a forward approximation for the derivative of F with respect to t at time $i \times \Delta T$ is given by

$$\frac{\partial F}{\partial t} \approx \frac{(f(i+1,j) - f(i,j))}{\Delta T}. \tag{18}$$

A central approximation for the derivative of F with respect to S at time $i \times \Delta T$ and stock price $j \times \Delta S$ is

$$\frac{\partial F}{\partial S} \approx \frac{(f(i,j+1) - f(i,j-1))}{2\Delta S}. \tag{19}$$

For the second derivative of F with respect to S at time $i \times \Delta T$ with stock price $j \times \Delta S$, we use a standard approximation

$$\frac{\partial^2 F}{\partial S^2} \approx \frac{(f(i,j+1) + f(i,j-1) - 2f(i,j))}{(\Delta S^2)}. \tag{20}$$

Now that we have the approximate derivatives, we can substitute these into the PDE (11), which leads to the following equation

$$\begin{aligned} 0 = & \frac{f(i+1,j) - f(i,j)}{\Delta T} + (r-y) \times (j\Delta S)\left(\frac{f(i,j+1) - f(i,j-1)}{2 \times \Delta S}\right) \\ & + \frac{1}{2}\sigma^2 (j\Delta S)^2 \left(\frac{f(i,j+1) + f(i,j-1) - 2f(i,j)}{(\Delta S)^2}\right) - rf(i,j). \end{aligned} \tag{21}$$

We may rewrite the equation as

$$f(i,j-1) \cdot a(j) + f(i,j) \cdot b(j) + f(i,j+1) \cdot c(j) = f(i+1,j), \tag{22}$$

and the coefficients a, b and c are defined by

$$a(j) = \frac{1}{2}(r-y) \times j\Delta T - \frac{1}{2}\Delta T \sigma^2 j^2, \tag{23}$$

$$b(j) = 1 + \sigma^2 j^2 \Delta T + r\Delta T, \tag{24}$$

and

$$c(j) = -\frac{1}{2}(r-y) \times j\Delta T - \frac{1}{2}\Delta T \sigma^2 j^2 \tag{25}$$

for $i = 0, 1, 2, ..., N-1$ and $j = 0, 1, 2, ..., M-1$. Note that these values will vary by the steps j of the stock price and not vary by i steps of time. A graphical representation of equation

(22) is given in Figure (5)[4] below.

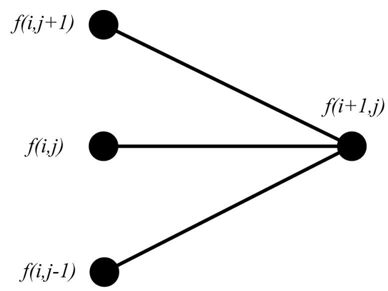

Figure 5: Graphical Representation of Equation (22)

The implicit finite difference method is known to be stable under some assumptions about the coefficients. A lemma is used to demonstrate the stability of the implicit finite difference method.

Lemma 1[5] *If* $b(j) \geq 0$, $a(j) \leq 0$ *and* $c(j) \leq 0$, $i = 0, 1, .., (N-1)$, $j = 1, .., (N-1)$, *then the implicit method is stable.*

Proof: Suppose $f(i,j)$ and $\tilde{f}(i,j)$ both satisfy the same boundary conditions and the equation (22) for some $i \in \{0, 1, ..., N-1\}$ and that $\mid f(i+1,j) - \tilde{f}(i+1,j) \mid \leq \varepsilon \forall j$. Denote

$$E_j = f(i,j) - \tilde{f}(i,j),\ j = 0, .., M.$$

Denote by V the maximal value of $\mid E_j \mid$, $j = 0, ..., M$. We want to show that $V \leq \varepsilon$; this shows the stability of the system. Since both $f(i,j)$ and $\tilde{f}(i,j)$ satisfy (22), their difference also satisfies the system. We write the equation for the difference in the form

$$b(j)E_j = f(i+1,j) - \tilde{f}(i+1,j) - a(j)E_{j-1} - c(j)E_{j+1}.$$

By taking absolute values of both sides and using properties of the absolute value, we get

$$b(j) \mid E_j \mid \leq \varepsilon - a(j) \mid E_{j-1} \mid - c(j) \mid E_{j+1} \mid .$$

Here we used all of the assumptions of the lemma. We can make the right hand side larger, by replacing the absolute values of E_{j-1} and E_{j+1} with the maximal value V:

$$b(j) \mid E_j \mid \leq \varepsilon - a(j)V - c(j)V.$$

[4] Made in Adobe Illustrator CS3 by Kathrine Salamonsen. Source of illustration: http://www.goddardconsulting.ca/option-pricing-finite-diff-implicit.html.

[5] See Section 1.7.2 The stability of the basic implicit method p.30-31 Computational Finance (2011) by Raul Kangro for the origin of the Lemma.

The last inequality holds for all $j = 1, .., M-1$. Choose the value of $j \in \{1, .., M-1\}$ such that $\mid E_j \mid = V$. In the case of that j we have

$$b(j)V \leq \varepsilon - a(j)V - c(j)V$$

hence

$$(a(j) + b(j) + c(j))V \leq \varepsilon.$$

But $a(j) + b(j) + c(j) = 1 + r\Delta T$, hence we have shown that

$$V \leq \frac{\varepsilon}{1 + r\Delta t} < \varepsilon.$$

This proves the lemma.

As the solution procedure work backwards, from right to left, it is called the implicit finite difference method. At each time point, we have a set of simultaneous equations to solve. In order to value the option at $t = 0$, we start backwards by finding the payoff at maturity which are the known values $f(i, j)$ for $i = N$. One time step before maturity at $i = N - 1$, the boundary conditions gives values at $j = 0$ and M. For $j = 1, 2, 3, .., M - 1$ the values $f(N - 1, j)$ are still unknown. As we consider $N = 3$ and $M = 6$ the process to solve the set of simultaneous equation are given to correspond. In equation (22), for $N = 3$, $i = 2$, this will be the starting point of solving the equations. Boundaries are excluded, thus j varies from 1 to 5, and our set of equations are

$$f(2,0) \cdot a(1) + f(2,1) \cdot b(1) + f(2,2) \cdot c(1) = f(3,1), \tag{26}$$

$$f(2,1) \cdot a(2) + f(2,2) \cdot b(2) + f(2,3) \cdot c(2) = f(3,2), \tag{27}$$

$$f(2,2) \cdot a(3) + f(2,3) \cdot b(3) + f(2,4) \cdot c(3) = f(3,3), \tag{28}$$

$$f(2,3) \cdot a(4) + f(2,4) \cdot b(4) + f(2,5) \cdot c(4) = f(3,4), \tag{29}$$

and

$$f(2,4) \cdot a(5) + f(2,5) \cdot b(5) + f(2,6) \cdot c(5) = f(3,5). \tag{30}$$

We may rewrite the first equation (26) as

$$f(2,1) \cdot b(1) + f(2,2) \cdot c(1) = f(3,1) - f(2,0) \cdot a(1). \tag{31}$$

Values on the RHS are known from the boundary conditions, while the values on the LHS still are unknown. Equations (27) - (29) in the middle can be left the way they are, but the last equation needs to be adjusted as

$$f(2,4) \cdot a(5) + f(2,5) \cdot b(5) = f(3,5) - f(2,6) \cdot c(5). \tag{32}$$

Now we have a set of five simultaneous equations (31), (27), (28), (29) and (32) in five unknowns, we can express these in matrix form as

$$\begin{pmatrix} b(1) & c(1) & 0 & 0 & 0 \\ a(2) & b(2) & c(2) & 0 & 0 \\ 0 & a(3) & b(3) & c(3) & 0 \\ 0 & 0 & a(4) & b(4) & c(4) \\ 0 & 0 & 0 & a(5) & b(5) \end{pmatrix} \times \begin{pmatrix} f(2,1) \\ f(2,2) \\ f(2,3) \\ f(2,4) \\ f(2,5) \end{pmatrix} = \begin{pmatrix} f(3,1) \\ f(3,2) \\ f(3,3) \\ f(3,4) \\ f(3,5) \end{pmatrix} - \begin{pmatrix} f(2,0)a(1) \\ 0 \\ 0 \\ 0 \\ f(2,6)c(5) \end{pmatrix}. \tag{33}$$

$$\mathbf{A} \times \mathbf{f_i} = (\mathbf{f_{i+1}} - \mathbf{d_i})$$

where $\mathbf{A}$ is an $(M-1)\times(N-1)$ tridiagonal square matrix, and both $\mathbf{f}_i$ and $\mathbf{d}_i$ are vectors of dimension $M-1$. Our boundary conditions gives us the values for $f(3,\cdot)$, $f(2,0)$ and $f(2,6)$. The unknown values remains to find for $f(2,\cdot)$. If we rearrange the equation, we can solve for these unknown variables

$$\begin{pmatrix} f(2,1) \\ f(2,2) \\ f(2,3) \\ f(2,4) \\ f(2,5) \end{pmatrix} = \begin{pmatrix} b(1) & c(1) & 0 & 0 & 0 \\ a(2) & b(2) & c(2) & 0 & 0 \\ 0 & a(3) & b(3) & c(3) & 0 \\ 0 & 0 & a(4) & b(4) & c(4) \\ 0 & 0 & 0 & a(5) & b(5) \end{pmatrix}^{-1} \times \begin{pmatrix} f(3,1) - f(2,0)a(1) \\ f(3,2) \\ f(3,3) \\ f(3,4) \\ f(3,5) - f(2,6)c(5) \end{pmatrix} \tag{34}$$

a representation of the general solution is $\mathbf{f}_i = \mathbf{A}^{-1} \times (\mathbf{f}_{i+1} - \mathbf{d}_i)$, which gives the vector of option values at time step i in terms of those at time step $i+1$. Applying the equation (34) by using backward recursion from time step $i = N-1$ to $i = 0$ we eventually will find the option value at time 0. Option values at $i = 2$, are found by multiplying the inverse matrix with the values of the option at $t = 3$ in the grid.

The solution in equation (34) provides us with the values of the function f at time $i = 2$ in terms of f values of $i = 3$. In general, we can obtain the values of $f(i-1,\cdot)$ from the values $f(i,\cdot)$. To find the values for $i = 1$ and $i = 0$, we use the exact same process, by using $f(2-1,\cdot)$ and $f(1-1,\cdot)$ from the values of $f(3,\cdot)$. Option values at the beginning of the grid $f(0,\cdot)$ is at the end calculated from the known maturity values at $f(T,\cdot)$. As N and M get bigger then the finite difference method will converge to the correct option value at time 0.

3.3 Least-Square Monte Carlo Method

Longstaff and Schwartz (2001) introduced the least squares Monte Carlo method of valuing American options. Their process starts by generating a chosen number ω of price path for a stock over a chosen number of time periods T. Each stock price path will be different when we use simulation of the paths.

The objective of the least squares approach is to find the optimal stopping time that maximizes the value of the American option. By using least squares regression, it is possible to estimate a conditional expectation function that gives an expected value of continuing to hold the option. By comparing the value of immediate exercise with the expected value of continuing to hold the option, one can find when the option should be exercised. After finding the optimal stopping time for the chosen number of paths ω, the optimal payoff for each path is discounted back to $t = 0$. At the end, the value of the American put option is found by averaging the discounted payoff by the number of paths. The advantage of this simulation method as opposed to finite difference and the binomial method, is that it is simple to apply when the value of the option depends on multiple factors. In this section, the valuation framework and the notation necessary to describe the LSM algorithm is presented.

3.3.1 Valuation Framework for Least Squares Monte Carlo

Let S be the stock price at time t. The expected drift in S is assumed to be μS, where μ is a constant parameter. μ is the expected rate of return on the stock. In a short interval of time, denoted as dt, the expected increase in S is $\mu S dt$. The volatility of a stock is represented by σ, which is the measure of uncertainty about the return on the stock. In risk-neutral pricing we change the parameter μ to r. Hence, the stock price is following the risk neutral stochastic differential equation

$$dS = rSdt + \sigma SdZ,$$

where r and σ are constants, and Z is a standard Brownian motion. The stock does not pay any dividends. The process to generate the stock price path in the risk-neutral world, and the solution to the stochastic differential equation (SDE), is given by

$$S_t = S_0 e^{(r-\frac{1}{2}\sigma^2)\delta t + \sigma\sqrt{\delta t} Z}$$

where Z is a geometric Brownian motion. The random variable S_t has a lognormal distribution, its log is normally distributed.

The least squares approach assume a complete probability space$(\Omega, \mathcal{F}, P)$ and a finite time horizon $[0, T]$. The probability space consist of Ω, the set of all possible realization and usually has an element ω sample path, $\mathcal{F}$ is the filtration element that represents information until time T. P is the probability measure defined on the elements of $\mathcal{F}$. Q is the martingale measure consistent with the no-arbitrage argument. We are interested in valuing American options with random cash flows which may occur over$[0, T]$. We consider options with payoffs that are elements of the space of square-integrable or finite-variance functions $L^2(\Omega, \mathcal{F}, Q)$. The path of cash flows generated by the option defined by $C(\omega, s; t, T)$, conditional on the option not being exercised at or before time t, and the optionholder follows the optimal stopping strategy for all s, $t < s \leq T$.

The LSM method aims to give a pathwise approximation to the optimal stopping rule that maximizes the value of the option. Although, in practice options are continuously exercisable, we use discrete exercise times in this framework. Consider K discrete times $0 < t_1 \leq t_2 \leq t_3 \leq ... \leq t_K = T$, where the option is exercisable and we examine the optimal stopping strategy at each exercise date. The investor has to decide wether to exercise the option or let it expire at the final expiration date. He chooses to exercise the option if it is in the money, or let it expire if it is out of the money. Prior to the final expiration date t_k, the optionholder must decide if he wants to exercise the option immediately, or let the option continue and make a decision at the next date wether he should exercise or not. The optionholder decides to exercise the option as soon as the value of immediate exercise is larger or equal to the value of continuation. However at time t_k, the cash flows from continuation are not known. Hence, the value of continuation is given by taking the expectation of the remaining discounted cash flows $C(\omega, s; t_k, T)$ with respect to the risk-neutral pricing measure Q. Thus, at time $t_{k,}$ we express the value of continuation $F(\omega; t_k)$ as

$$F\left(\omega; t_k\right) = E_Q\left[\sum_{j=k+1}^{K} \exp\left(-\int_{t_k}^{t_j} r(\omega, s) ds\right) C(\omega, t_j; t_k, T) \mid \mathcal{F}_{t_k}\right] \tag{35}$$

$r(\omega, t)$ is the riskless discount rate and $C(\omega, t_j; t_k, T)$ is the discounted cash flow. The expectation is taken conditional on the information set$\mathcal{F}_{t_k}$ at time t_k. By this representation, the problem of optimal exercise reduces to compare the value of the conditional expectation with the value of immediate exercise, and then exercise when the value of immediate exercise is larger or equal to the conditional expectation.

3.3.2 The LSM Algorithm

The conditional expectation function at time $t_{K-1}, t_{K-2}, ..., t_1$ is approximated by using least squares. The paths of cash flows $C(\omega, s; t, T)$ generated by the option is defined recursively, meaning we work backwards, since the cash flow at t_k can differ from the cash flow at t_{k-1}. This is because over a given timeframe it could be optimal to stop at an earlier date. At time t_{K-1}, the function $F(\omega; t_{K-1})$ can be represented as a linear combination of a countable set of $\mathcal{F}_{t_{K-1}}$- measurable basis functions. If the conditional expectation function is an element of the L^2 space of square-integrable functions and because L^2 is a Hilbert space with countable orthonormal basis, the conditional expectation can be represented as a linear function of the elements of the basis. One option is to use the set of weighted Laguerre polynomials as basis functions. Longstaff and Schwartz suggested to use

$$L_0(X) = \exp(-X/2), \tag{36}$$

$$L_1(X) = \exp(-X/2)(1 - X), \tag{37}$$

$$L_2(X) = \exp(-X/2)(1 - 2X + X^2/2), \tag{38}$$

$$L_n(X) = \exp(-X/2)\frac{e^X}{n!}\frac{d^n}{dX^n}(X^n e^{-X}), \tag{39}$$

as basis functions. Hermite, Legendre, Chebyshev, Gegenbauer and Jacobi are other types of basis functions that could be used. Now, the value of continuation $F(\omega; t_{K-1})$ can be represented as

$$F(\omega; t_{K-1}) = \sum_{j=0}^{\infty} a_j L_j(X). \tag{40}$$

Here we assume X is the value of the asset underlying the option and that X follows a Markov process. a_j coefficients are constants. The LSM approach is implemented by approximating $F(\omega; t_{K-1})$ using the first $M < \infty$ basis function. Denote this approximation as $F_M(\omega; t_{K-1})$. Next, $F_M(\omega; t_{K-1})$ is estimated by regressing the discounted values of $C(\omega, s; t_{K-1}, T)$ onto the basis functions for the paths where the option is in the money at time t_{K-1}. By limiting the region of where the conditional expectation function is estimated, less basis functions are needed to obtain an accurate approximation to the conditional expectation function. We use the three first basis functions (36), (37) and (38) in our analysis. Theorem 3.5 of White (1984) is used to show that the fitted value of the regression $\widehat{F_M}(\omega; t_{K-1})$ converges to $F_M(\omega; t_{K-1})$ as the number N of in the money paths in the simulation goes to infinity.

After estimating the conditional expectation function at time t_{K-1}, we are able to decide wether early exercise at time t_{K-1} is optimal for an in the money path ω by comparing the value of immediate exercise with $\widehat{F_M}(\omega; t_{K-1})$, and repeating this process for each in the money path. Once we have identified the exercise decision, then we can approximate the cash flow paths $C(\omega, s; t_{K-2}, T)$ from the option. Continue to repeat this process for each price path until the exercise decision at each time has been made. At last, the American option is valued by starting at time zero, moving forward along each path until the first stopping time occurs, and discount the cash flow from exercise back to time zero, then taking the average over all paths ω.

4 Implementation of Valuing American Options

4.1 Implementation of the Binomial Method in Excel

A spreadsheet found at http://investexcel.net/binomial-tree-american-option/ can be used to value the American put option by the binomial method. The goal is to find the value of the option today by working recursively, and determine when the option should be exercised or if we should keep the option for one more period. Table (1) contains the parameters with their values needed to value the American put option in the binomial method. By inserting these values into the excel spreadsheet one can obtain the lattice tree shown in Figure (6).

Parameter	**Meaning**	**Value**
n	Number of Nodes	3
T	Time to Maturity	1
r	Risk-free Rate	5%
σ	Volatility	32%
S_0	Initial Stock Price	1.0
K	Strike Price	1.05
u	Upward Movement in Stock Price	1.2029
d	Downward Movement in Stock Price	0.8313
p	The Risk-Neutral Probability	0.5

Table 1: Parameters Used in the Binomial Method

The possible stock prices is shown in the yellow boxes. Recall Figure (2) how the stock price tree is constructed. The stock price is starting at an initial price of $S_0 = 1.0$. u and d are found by Equation (5) and (6). By multiplying the initial price with $u = 1.2029$ the stock price move up one step. For a downward movement, multiplying the initial price with $d = 0.8313$ the stock price will move down one step. To do create a stock price tree by hand, one can follow the structure in Figure (2). Once the stock price tree has been made, the next step is to find the payoff at expiry. Recall the structure of the option price tree in Figure (3). As an example, at expiry $n = 3$, the payoff for Vud^2 is $Vud^2 = max(K - Sd^2, 0) = max(1.05 - 0.83131, 0) = 0.21869$. After the payoff at expiry for each node has been found, we compare the value of exercising the option with the value keeping the option. For example the value of exercising the option at $n = 2$ is $Vd^{2\,exercise} = max(K - S_0d^2, 0) = max(1.05 - 0.6910, 0) = 0.3589$ and the value of keeping the option is $Vd^{2\,keep} = e^{-r\delta t}[pVd^2u + (1-p)Vd^3] = e^{-0.05 \cdot 1/3}[0.5 \cdot 0.2186 + 0.5 \cdot 0.4755] = 0.3413$. At this node, the optimal strategy is to exercise the option as the value of exercising is larger than keeping the option.

The price and value of the option today is $V_0 = e^{-r\delta t}[pVu + (1-p)Vd] = e^{-0.05 \cdot 1/3}[0.5 \cdot 0.0530 + 0.50.2296] = 0.1391$. Immediate exercise is $V_0 = max(K - S, 0) = max(1.05 - 1.00, 0) = 0.05$.

0	1	2	3
			1,740646
			0
		1,447017	
		0	
	1,20292		1,20292
	0,053056		0
1		1	
0,13917		0,107716	
	0,83131		0,83131
	0,229669		0,21869
		0,691077	
		0,358923	
			0,574499
			0,475501

Figure 6: Lattice Tree of Possible Stock Prices and Option Values

To find the value of the option today, it is also possible to use Matlab. The program to run in Matlab can be found in the Appendix (7.3). By running this program, it will only give the value of the option today, it will not give the value step by step as in the Excel example.

4.2 Implementation of the Implicit Finite Difference Method in Excel

This section gives an implementation of the implicit finite difference method in a Norwegian version of Excel. This implementation is based on the Excel implementation of the implicit finite difference method for option pricing by Kyng, Purcal and Zhang (2016). The goal is to obtain the price of the American put option.

Figure (7) contains the parameters used with the corresponding value chosen to value the American put option in the implicit finite difference method. The parameter values ranges from cell A6:A18, and insert the values according to Figure (7) in a Excel spreadsheet.

	A	B
1	**Implicit finite difference method spread sheet**	
2		
3	**Valuation of an American put option**	
4		
5	**Table 1: Parameter values for the numerical example**	
6	1	S=initial stock price
7	1,05	X=exercise price for put
8	1	T=term to maturity
9	3	N=number of time increments (steps)
10	6	M=number of stock price increments
11	2	Smax=maximum stock price
12	0	Smin=minimum stock price
13	2	Range=Smax-Smin
14	0,333333333	Delta S
15	1/3	Delta T
16	0,05	Risk free interst rate
17	0	Dividend yield
18	0,32	Volatility
19		

Figure 7: Parameter Values

To generate the stock price grid called Table 2 in Excel corresponding to Figure (8), we insert in cell F7 =$A$12+$E7*$A$14, and copy this to cell I13 in order to obtain all the possible stock prices. Stock prices are generated according to $S_{(i,j)} = j \times \Delta S$ at time $i \times \Delta T$. i indexes time in the columns and j indexes the stock price in the rows.

Table 2: Stock price S(i,j) at all nodes on the grid corresponding to the parameters

	0	1	2	3
0	0,0000	0,0000	0,0000	0,0000
1	0,3333	0,3333	0,3333	0,3333
2	0,6667	0,6667	0,6667	0,6667
3	1,0000	1,0000	1,0000	1,0000
4	1,3333	1,3333	1,3333	1,3333
5	1,6667	1,6667	1,6667	1,6667
6	2,0000	2,0000	2,0000	2,0000

Figure 8: Stock Price Grid

We use the boundary condition equations (12),(16) and (17) to calculate the values of $f(i,j)$ on the boundary of the grid. The values on the righthand side are calculated according to the boundary equation (12) at maturity. The excel code in cell I19 is =STØRST($A$7-I7;0) and copy this to cell I25. The bottom of the grid is calculated according to equation (16), here we insert zero in cell F25:H25. The upper row is calculated according to boundary condition (17) and the excel code in cell F19 is given by =$A$7*EKSP(-($A$9-F$18)*$A$15*$A$16)-F7*EKSP(-($A$9-F$18)*$A$15*$A$17) and copied to cell H19. Figure (9) shows the values along the boundary of the grid.

Table 3: Values of f(i,j) along the boundary of the grid

	0	1	2	3
0	0,99879	1,01558	1,03265	1,05000
1				0,71667
2				0,38333
3				0,05000
4				0,00000
5				0,00000
6	0,00000	0,00000	0,00000	0,00000

Figure 9: Values Along the Boundary of the Grid

The implicit coefficients are calculated according to the equation (23), (24) and (25) by inserting the parameter values from table (7). Figure (10) shows the values obtained for the coefficients. To compute the $a(j)$ coefficient we insert =0,5*($A$16-$A$17)*E31*$A$15-0,5*$A$18^2*E31^2*$A$15 in cell F31 and copy to cell F37.

In cell G31 we insert =1+$A$18^2*E31^2*$A$15+$A$16*$A$15 and copy to cell G37 to compute the $b(j)$ coefficients. For the last coefficient $c(j)$, we insert =-0,5*($A$16-$A$17)*E31*$A$15-0,5*$A$18^2*E31^2*$A$15 in cell H31 and copy to cell H37. Observe that the stability condition for the coefficients is met, and Lemma 1 holds,$a(j)+b(j)+c(j) = 1 + r\Delta T$.

Table 4: coeffecients for our implicit finite difference example

	a(j)	b(j)	c(j)
0	0,0000	1,0167	0,0000
1	-0,0087	1,0508	-0,0254
2	-0,0516	1,1532	-0,0849
3	-0,1286	1,3239	-0,1786
4	-0,2397	1,5628	-0,3064
5	-0,3850	1,8700	-0,4683
6	-0,5644	2,2455	-0,6644

Figure 10: Coefficients $a(j)$, $b(j)$ and $c(j)$

The values for the tridiagonal matrix A is presented in Figure (11). To obtain the values for the tridiagonal matrix A, it is best to copy the code in cell G32 and insert it into F42 and then copy. By copying the codes into the tridiagonal matrix, it will be easier if one wants to change some of the parameter values for other examples. It is also possible to just enter the numbers from Figure (10) but then it will only work for this numerical example.

Table 5: Tri-diagonal matrix, A

	1	2	3	4	5
1	1,0508	-0,0254	0,0000	0,0000	0,0000
2	-0,0516	1,1532	-0,0849	0,0000	0,0000
3	0,0000	-0,1286	1,3239	-0,1786	0,0000
4	0,0000	0,0000	-0,2397	1,5628	-0,3064
5	0,0000	0,0000	0,0000	-0,3850	1,8700

Figure 11: Tridiagonal Matrix A

Figure (12) contains the values for the adjustment vector. In cell F50 we calculate the entries of the adjustment vector by entering the code =F19*$F$32 and copy to cell H50. Enter =F25*$H$36 in cell F54 and copy to H54.

	C	D	E	F	G	H	I
47							
48		**Table 6: Adjustment vector, d_i**					
49				**0**	**1**	**2**	
50			**1**	-0,008722774	-0,008869372	-0,009018433	
51			**2**	0	0	0	
52			**3**	0	0	0	
53			**4**	0	0	0	
54			**5**	0	0	0	
55							

Figure 12: Adjustment Vector d_i

To value the American put option we first enter =STØRST($A$7-I8;0) in cell I59 and copy to I63. Then enter =STØRST(INDEKS(MMULT(MINVERS($F$42:$J$46);(G$59:G$63-F$50:F$54));$E59); STØRST($A$7-F8;0)) in cell F59 and copy to H63 to compute the rest of the option prices. Finally the American put option values are presented in Figure (13). The American put option price is given in the yellow box $f(0,3) = 0.11336$.

	C	D	E	F	G	H	I	J
56								
57		**Table 7: Option Prices**						
58				**0**	**1**	**2**	**3**	
59			**1**	0,71667	0,71667	0,71667	0,71667	
60			**2**	0,38333	0,38333	0,38333	0,38333	
61			**3**	0,11336	0,09606	0,07526	0,05000	
62			**4**	0,03457	0,02365	0,01203	0,00000	
63			**5**	0,01043	0,00619	0,00248	0,00000	
64								

Figure 13: Option Prices for the American Put Option

4.3 Implementation of Least Square Monte Carlo Method in STATA

This section is based on Longstaff and Schwartz (2001) numerical example. We consider an American put option on a share of a non-divided paying stock, with a strike price $K = 1.05$, which is exercisable at time $t = 1$, 2 and 3. The risk-less rate is set to $r = 5\%$ and volatility $\sigma = 0.32$. $\Delta t = 1/3$. To present the implementation from the algorithm we choose only eight paths for simplicity. These stock price paths are generated under the risk-neutral measure Q. These stock price paths can be generated in STATA by the running the program in Appendix (7.1). Eight simulated stock price paths are presented in the following matrix

Path	$t = 0$	$t = 1$	$t = 2$	$t = 3$
1	1.00	1.14	0.91	0.79
2	1.00	1.14	1.67	1.73
3	1.00	0.74	0.71	0.88
4	1.00	0.97	0.84	0.89
5	1.00	1.30	1.13	1.32
6	1.00	1.02	0.94	0.73
7	1.00	0.89	0.81	1.03
8	1.00	1.01	1.04	1.11

Table 2: Stock Price Paths

After all the stock price paths has been generated, the values obtained are entered in the data editor in STATA. Name each variable $t0$, $t1$, $t2$ and $t3$ as shown below

	t0	t1	t2	t3
1	1	1.14	.91	.79
2	1	1.14	1.67	1.73
3	1	.74	.71	.88
4	1	.97	.84	.89
5	1	1.3	1.13	1.32
6	1	1.02	.94	.73
7	1	.89	.81	1.03
8	1	1.01	1.04	1.11

Figure 14: Stock Price Paths in STATA

and enter the stock prices according to table (2). The program given in Appendix (7.2) is a do-file for STATA that values the American option in the LSM method. As we continue the numerical example, the implementation of the least squares Monte Carlo method in STATA for an American put option is given along.

Our goal is to solve for the stopping rule that maximizes the value of the option at each point along each path. The algorithm is recursive, and we start considering the last time period for the option. Following the optimal strategy at time 3, the cash flow realized by the optionholder, conditional on not exercise the option before maturity, are presented in the cash-flow matrix

Path	$t = 1$	$t = 2$	$t = 3$
1	-	-	.26
2	-	-	.00
3	-	-	.17
4	-	-	.16
5	-	-	.00
6	-	-	.32
7	-	-	.02
8	-	-	.00

Table 3: Cash-Flow Matrix at Time 3

This is the cash flow a holder of a European option would get if the option were European instead of an American option. Discounting back the cash flow three periods and average over all paths gives us the European value of the option, which is 0.1000.

The commands in STATA are shown below. As the method is recursive, we consider time 3 first. First we generate the strike price sp and a variable for the cash flow at time 3 P3. The replace command sorts out in-the-money paths. d0 generates the cash flow matrix at time 3, by discounting the cash flow back to time zero. The summarize command take the average over all paths which gives the value of the European put option as shown in Figure (15).

```
// ---------------------- t = 3 ---------------------------------------
gen sp=1.05
gen P3=sp-t3
replace P3=0 if P3<0
gen d0=exp(-0.05*3)*P3
summarize d0            // the mean is the value of the EU option
```

```
. summarize d0          // the mean is the value of the EU option

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          d0 |         8    .1000573    .1111744          0   .2754265
```

Figure 15: STATA: Value of the European Put Option

At time 2 the optionholder must decide if he want to exercise the option immediately or continue to hold the option until the final expiration date time 3. Let X denote the stock price where the option is in-the-money at time 2. The matrix presents six possible paths for the stock prices where the option is in-the-money. Let Y denote the discounted cash flow received at time 3 if the option is not exercised at time 2. The conditional expectation function is better estimated when we consider only in-the-money paths for the stock. The efficiency of the algorithm is also significantly improved. Vector X and Y are given in the matrix below

Path	Y	X
1	$.26\times.95122$	.91
2	-	-
3	$.17\times.95122$	.71
4	$.16\times.95122$	.84
5	-	-
6	$.32\times.95122$	.94
7	$.02\times.95122$	.81
8	$.00\times.95122$	1.04

Table 4: Regression at Time 2

Regressing Y on a constant X and X^2, will give an estimate for the expected cash flow from continuing to hold the option, conditional on the stock price at time 2. We obtained the conditional expectation function $E[Y \mid X] = -2.937 + 7.228X - 4.172X^2$. By inserting the value of X, where the stock price is in-the-money at time 2, into the conditional expectation function, we get an expected value of continuing to hold the option. The value of immediate exercise is equal to the intrinsic value, $1.05 - X$. In the first column the value of immediate exercise is given, and the expected value of continuation is given in the second column. We compare these two values for each path, and exercise the option when immediate exercise is higher or equal to the value of continuation.

Path	Exercise	Continuation
1	.14	.1856
2	-	-
3	.34	.0917
4	.21	.1907
5	-	-
6	.11	.1709
7	.24	.1804
8	.01	.0676

Table 5: Optimal Early Exercise Decision at Time 2

This implies that it is optimal to exercise the option at time 2 for the third, fourth and seventh path. The following matrix present the cash flow received from exercising and the cash flow from continuing to hold the option.

Path	$t=1$	$t=2$	$t=3$
1	-	.00	.26
2	-	.00	.00
3	-	.34	.00
4	-	.21	.00
5	-	.00	.00
6	-	.00	.32
7	-	.24	.00
8	-	.00	.00

Table 6: Cash-Flow Matrix at Time 2

Observe that when the option is exercised at time 2, column for time 3 becomes zero. This is because the option can only be exercised once and there will be no further cash flows. Future cash flow can only occur at time 2 or time 3, but not both.

Below are the commands in STATA considered at time 2. First we generate a new variable xt2 containing the stock prices at time 2. The replace command eliminates the cases where the option is out-of-the-money. Then we generate a variable y2 which is the discounted payoff at time 2. We need to generate the variable X^2 called xt22, which is the stock price squared. The regress command is estimating the conditional expectation function at time 2 by polynomial regression. Figure (16) shows the estimated coefficients for the conditional expectation function at time 2, $E[Y \mid X] = -2.937 + 7.228X - 4.172X^2$. Then the immediate exercise value is obtained by generating the variable ex2. The replace command sorts out the cases where the option is being exercised. The cash flow matrix at time 2 will be made by generating the variable P2. At the end, the replace command is updating the cash flow matrix at time 2 to be the same as in (6).

```
// ----------------------- t = 2 ----------------------------------------
gen xt2 = t2
replace xt2=. if xt2>sp    // eliminate out-of-money cases
gen y2=exp(-0.05*1)*P3
gen xt22=xt2^2
regress y2 xt2 xt22        // estimate polynomial regression
predict y2hat, xb
gen ex2=sp-xt2
replace ex2=0 if ex2<y2hat
```

```
gen P2=ex2                // set up P2
replace P2=0 if ex2==.
replace P3=0 if P2>0      // update P3
```

```
. regress y2 xt2 xt22        // estimate polynomial regression

      Source |       SS       df       MS              Number of obs =       6
-------------+------------------------------           F(  2,     3) =    0.37
       Model |  .014442719     2  .00722136            Prob > F      =  0.7187
    Residual |  .058622871     3  .019540957           R-squared     =  0.1977
-------------+------------------------------           Adj R-squared = -0.3372
       Total |  .07306559      5  .014613118           Root MSE      =  .13979

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         xt2 |   7.228185   8.616349     0.84   0.463    -20.19288    34.64925
        xt22 |  -4.172396   4.913703    -0.85   0.458    -19.80999     11.4652
       _cons |  -2.937287    3.73971    -0.79   0.490    -14.83871    8.964138
------------------------------------------------------------------------------
```

Figure 16: STATA: Regression at Time 2

Next, we analyze if the option should be exercise at time 1. From the stock price matrix, at time 1 there are five paths where the option is in-the-money. For these paths, we define X as the stock price at time 1, and Y as the discounted cash flow from time 2. The actual realized cash-flow along each path is used in defining Y. This is because discounting back the conditional expected value could lead to an upward bias in the value of the option.

The nondashed elements in the matrix shows the vectors for X and Y.

Path	Y	X
1	-	-
2	-	-
3	.34×.95122	.74
4	.21×.95122	.97
5	-	-
6	.00×.95122	1.02
7	.24×.95122	.89
8	.00×.95122	1.01

Table 7: Regression at Time 1

Again, we estimate the conditional expectation function at time 1, by regressing Y on a constant X and X^2. This gives the estimated conditional expectation function $E[Y \mid X] = -3.290 + 9.221X - 5.871X^2$.

By substituting the value of X into the estimated conditional function, we obtain the expected value of continuation. Column one represent the immediate exercise value and column two gives the estimated expected value of continuation. Comparing the two values, we see that we should exercise for the sixth and eighth paths.

Path	Exercise	Continuation
1	-	-
2	-	-
3	.31	.3185
4	.08	.1303
5	-	-
6	.03	.0072
7	.16	.2662
8	.04	.0342

Table 8: Optimal Early Exercise Decision at Time 1

The commands to do the valuation of the option at time 1 is given below. First we generate a variable xt1 which is the stock price at time 1. Then the cases where the option is out-of-the-money is eliminated. Generation of the variable y1 is the discounted cash flow at time 1 for in-the-money cases. Then we generate a new variable X^2, which is the stock price squared at time 1. Again, we use polynomial regression to obtain the conditional expectation function at time 1. Figure (17) shows the estimated coefficients for the conditional expectation function at time 1, $E[Y \mid X] = -3.290 + 9.221X - 5.871X^2$. ex1 gives the immediate exercise value and the replace command sorts out the cases where the option is being exercised. P1 gives the cash flow matrix at time 1 and the replace command updates the matrix for where it is optimal to exercise the option.

```
// ------------------------ t = 1 ----------------------------------------
gen xt1 = t1
replace xt1=. if xt1>sp    // eliminate out-of-money cases
gen y1=exp(-0.05*1)*P2
gen xt12=xt1^2
regress y1 xt1 xt12        // estimate polynomial regression
predict y1hat, xb
gen ex1=sp-xt1
replace ex1=0 if ex1<y1hat
gen P1=ex1
replace P1=0 if ex1==.
replace P2=0 if P1>0      // update P2
```

```
replace P3=0 if P1>0       // update P3
```

```
. regress y1 xt1 xt12        // estimate polynomial regression

      Source |       SS       df       MS              Number of obs =       5
-------------+------------------------------           F(  2,     2) =   10.16
       Model |  .07617896     2  .03808948             Prob > F      =  0.0896
    Residual | .007500391     2  .003750195            R-squared     =  0.9104
-------------+------------------------------           Adj R-squared =  0.8207
       Total |  .08367935     4  .020919838            Root MSE      =  .06124

------------------------------------------------------------------------------
          y1 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         xt1 |   9.221392   6.304011     1.46   0.281    -17.90258    36.34536
        xt12 |  -5.871555   3.579844    -1.64   0.243    -21.27438     9.53127
       _cons |  -3.290913   2.735395    -1.20   0.352    -15.06037     8.47854
------------------------------------------------------------------------------
```

Figure 17: STATA: Regression at Time 1

After identifying the optimal exercise strategy for time 1, 2 and 3, we are able to identify when we should exercise the option for each path. The number one indicates where the option should be exercised.

Path	$t = 1$	$t = 2$	$t = 3$
1	0	0	1
2	0	0	0
3	0	1	0
4	0	1	0
5	0	0	0
6	1	0	0
7	0	1	0
8	1	0	0

Table 9: Stopping Rule

From this specification of the stopping rule, is easy to determine the realized cash flow by following the stopping rule. Where there is a one in the matrix above, the option is exercised at that time. Below is the realized cash flow for each path along with the time it should be exercised.

Path	$t=1$	$t=2$	$t=3$
1	.00	.00	.26
2	.00	.00	.00
3	.00	.34	.00
4	.00	.21	.00
5	.00	.00	.00
6	.03	.00	.00
7	.00	.24	.00
8	.04	.00	.00

Table 10: Option Cash Flow Matrix

Discounting back the future cash flow for each path by e^{-rt} to time zero, and averaging over all paths one finds the value of the American put option. This gives an value of 0.1256 for the American put option. Compared with the value of 0.1000 for the European put option.

The final commands values the American option today. Three new variables are generated pd3, pd2 and pd1, which are the discounted cash flows at time 1, 2 and 3. Then we generate total which is the sum of these discounted cash flows. At the end, by the command summarize total, we obtain the value of the American put option as seen in Figure (18).

```
//------------------------ t = 0 ------------------------------------------
gen  pd3 =  exp(-0.05*3 )*P3
gen  pd2 =  exp(-0.05*2 )*P2
gen  pd1 =  exp(-0.05*1 )*P1
gen total = pd1 + pd2 + pd3
summarize total          // the mean gives the value of the AM option
//---------------------------------------------------------------------
```

```
. summarize total          // the mean gives the value of the AM option

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       total |         8    .1256489    .1218607          0   .3076447
```

Figure 18: STATA: Value of the American Put Option

Table (11) summarizes the value obtained for the American put option by using the three different numerical methods. The European put option values are also given to compare the value. American put options typically has a larger value than the European put due to the feature of early exercise as seen in the table. The reason for the lower value in the finite difference method, comes from the boundary condition. We only need to fix one boundary condition in the implicit finite difference method, that is for S_{max}, and can be difficult to choose. We set $S_{max} = 2 \cdot S = 2$ in this example. As $N \to \infty$ the boundary should be set higher. Conversely, for a shorter time period, the boundary should be deceased.

Numerical Method	**American Put**	**European Put**
Binomial Method	0.1391	0.0881
Finite Difference Method	0.1133	0.1099
Least Squares Monte Carlo	0.1256	0.1000

Table 11: Comparison of Results

4.4 Comparison of the Valuation Methods in Matlab

Table (12)[6] contains values for an American put option by using the three different valuation methods with a given set of parameters. We compare the difference of the value obtained for the American put option for the methods used. The idea behind the early exercise column, is to see how much more the American put option is worth because of its early exercise feature as opposed to the value of the European put option which can only be exercised at maturity. S is the initial stock price, which has five different values. The volatility σ has two different values. Number of years until maturity is given by T and has two values. The strike price $K = 100$. The risk-free rate is $r = 5\%$. Matlab was used to obtain the values in the table. To replicate the table, the programs in Appendix (7.3), (7.4) and (7.5) can be used. Although, one should note that the program for least square Monte Carlo will give different values in every attempt. This is because it is simulation and the stock price paths differ for every attempt. The program allowed for a maximum of $N = 50$ time steps and $M = 50$ stock price steps.

[6] Based on Longstaff and Schwartz's (2001) Table 1, Section 3 Valuing American Put Options, page 127.
Bin M (1) is the price of the American put option using the binomial method.
FDM (2) is the price of the American put using the finite difference method.
LSM (3) is the price of the American put using the least squares Monte Carlo method.
BS (4) is the price of a European put obtained the Black-Scholes formula.
(1)-(3) is the difference in price of the American put option obtained by the binomial method and least squares Monte Carlo method
(2)-(3) is the difference in price of the American put option obtained by the finite difference and least squares Monte Carlo method
(3)-(4) is the early exercise value showing how much more the American put option is worth compared with the European put option

S	σ	T	**Bin M (1)**	**FDM (2)**	**LSM (3)**	**BS (4)**	**(1)-(3)**	**(2)-(3)**	**Early exercise value (3)-(4)**
90	**.20**	**1**	11.4852	10.9195	12.6251	10.2141	-1.1399	-1.7056	2.4110
90	.20	2	12.6012	11.8183	11.7325	10.3925	0.8687	0.0858	1.3400
90	.40	1	18.1466	17.6936	18.2927	17.3726	-0.1461	-0.5991	.9201
90	.40	2	22.0408	21.3107	20.8924	20.3398	1.1484	0.4183	.5526
95	.20	1	8.4417	8.8481	9.9781	7.6338	-1.5364	-1.1300	2.3443
95	.20	2	9.9012	9.8015	11.6686	8.3250	-1.7674	-1.8671	3.3436
95	.40	1	15.8433	16.0600	15.5315	15.1306	0.3118	0.5285	.4009
95	.40	2	19.9546	19.7118	22.4999	18.4702	-2.5453	-2.7881	4.0297
100	.20	1	6.0737	7.0858	9.8146	5.5735	-3.7409	-2.7288	4.2411
100	.20	2	7.7011	8.0967	12.8670	6.6105	-5.1659	-4.7703	6.2565
100	.40	1	13.6257	14.5564	15.6318	13.1458	-2.0061	-1.0754	2.4860
100	.40	2	17.9393	18.2340	21.0994	16.7739	-3.1601	-2.8654	4.3255
105	.20	1	4.3120	5.0976	4.8299	3.9808	-0.5179	0.2677	.9219
105	.20	2	6.0290	6.3069	8.4242	5.2077	-2.3952	-2.1173	3.2165
105	.40	1	11.8889	12.6410	19.0821	11.3976	-7.1932	-6.4411	7.6845
105	.40	2	16.3389	16.5225	16.1370	15.2364	0.2019	0.3855	.9006
110	.20	1	3.0096	3.6014	4.4898	2.7858	-1.4802	-0.8884	1.7040
110	.20	2	4.6354	4.8904	5.7079	4.0738	-1.0725	-0.8175	1.6341
110	.40	1	10.2316	10.9597	10.1144	9.8642	0.1172	0.8453	.2502
110	.40	2	14.8420	14.9792	14.3669	13.8436	0.4751	0.6123	.5233

Table 12: Simulation of an American Put Option

In column 8 and 9 we observe that there are more negative values in the differences than positive difference. This means that the least squares Monte Carlo method has the tendency to slightly overvalue the option. One explanation for the differences could be the length of the time steps and/or the length of stock price steps, as all these methods has a convergence theorem, meaning that when $M \to \infty$ and $N \to \infty$ then the value will converge to the true value of the option.

The last column presents the difference in early exercise value. Here, we took the difference between the value in the least squares Monte Carlo for an American put option with early exercise feature with the value of an European put option that could only be exercised at maturity. Observe how much more value the American put option gives compared to the European.

Figure (19) illustrates a comparison of each valuation method of how fast the option price is being calculated in terms of time taken. To obtain these two graphs one can use the program given in the Appendix (7.6). The parameter numbers in bold in Table (12) were used for this comparison of the methods. The risk free rate is $r = 5\%$ and strike price is $K = 100$. Number of grid space and simulation paths were set to $M = 10000$. Maximum time steps were $N = 100$. The graph on the top in figure (19) illustrates the resulting price of the American option in terms of time taken. The price of the American put is 11.1797 for this simulation as seen on the $Y-$axis. The second plot shows the computational time taken for each method. The maximum time to compute is 0.3586 seconds for the LSM method, which is relatively quick. As for the binomial and finite difference the computation was even quicker.

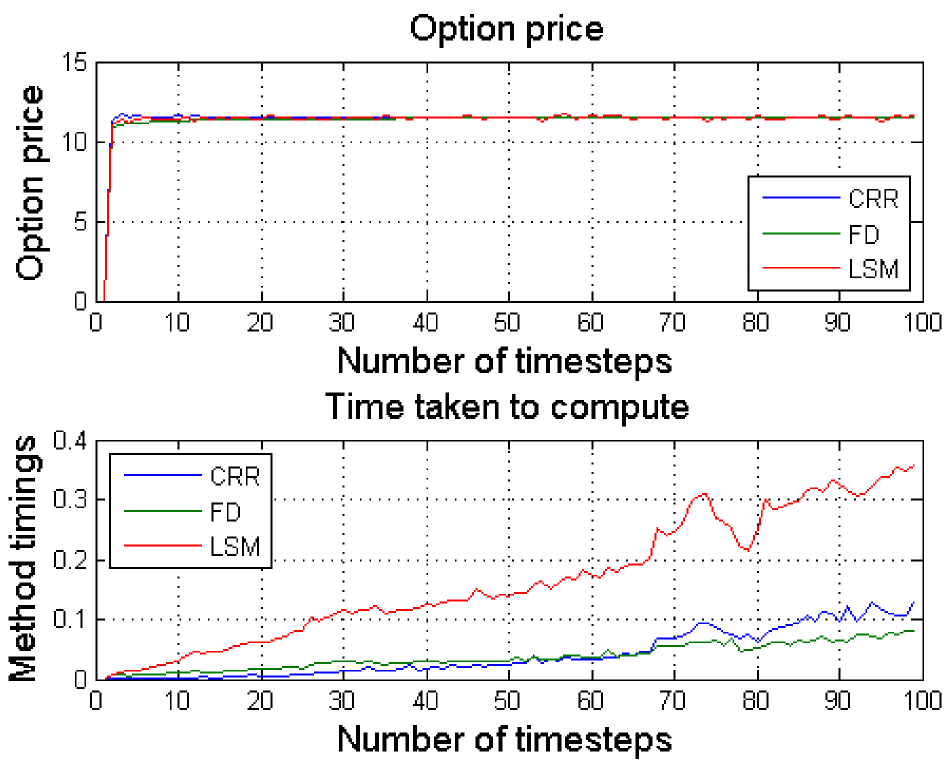

Figure 19: Computational Time

Figure (20) plots the comparison of each methods in a 3D surface. The option price is shown on the $Y-$axis ranging from 0 to 100. On the $X-$axis we have time to maturity ranging from 0 to 1. The steps of the stock price is shown on the $Z-$axis ranging from 0 to 800. The colored surface in the back is the finite difference method, the blue surface is the binomial method and the red and blurry dots in front is the least squares Monte Carlo. Each of the three surfaces are compared in this plot, and demonstrates the domain (minimum and maximum values) for the option price, time to maturity and the stock price steps. For example consider the blue surface for the binomial method. The maximum value of the stock price is 665 and the maximum value of the option price is 87.

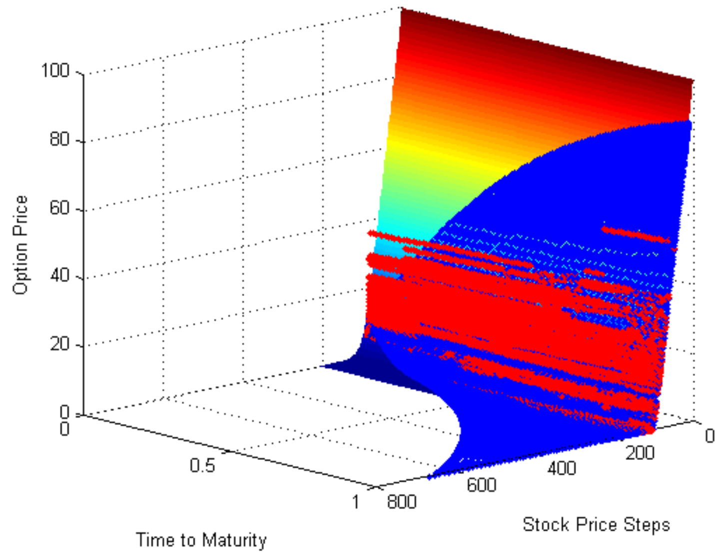

Figure 20: 3D Surface of the Valuation Methods

5 Conclusion

The goal of this thesis was to find the optimal stopping time for an American put option where the payoff is maximized. We introduced three possible valuation frameworks to value an American put option numerically. Explanation of how to implement each method for different softwares was given. First, we introduced the binomial method to find the optimal exercise strategy along the lattice three. Then we presented the implicit finite difference algorithm which approximates the partial differential equation by a difference equation. Implementation of the binomial method and the implicit finite difference method was given by a numerical examples in Excel. Last, we presented the least squares Monte Carlo approach to find the optimal stopping time maximizes the value of the American put option. We showed how to implement the least squares Monte Carlo approach by using STATA. At the end, we compared the valuation methods in terms of time taken to compute the value and looked at the differences in early exercise value of an American put option with the European put option value.

Suggested future research would be to include alternative basis function and check the performance by using the least squares Monte Carlo method. One could also try to estimate the conditional expectation function by using other alternative methods of ordinary least squares.

6 References

Black, Fisher; Scholes, Myron (1973): "The Pricing of Options and Corporate Liabilities". *Journal of Political Economy.* 81 (3): 637-654.

Broadie, Mark; Glasserman, Paul (1997): "Pricing American-style Securities Using Simulation", *Journal of Economic Dynamics and Control 21* (1997) 1323-1352.

Černý, Aleš (2009): "*Mathematical Techniques in Finance*" Tools for Incomplete Markets, 2nd Edition. Princeton University Press.

Chen, CW., Huang, KL. & Lyuu, YD. (2015):" Accelerating the Least-Square Monte Carlo Method with Parallel Computing". *The Journal of Supercomputing* Volume 71, Issue 9, pp 3593-3608.

Cox, John C.; Ross, Stephen A.; Rubinstein, Mark (1979): "Option Pricing: A Simplified Approach", *Journal of Financial Economics 7* (1979) 229-263.

Higham J. Desmond (2004): "*An Introduction to Financial Option Valuation, Mathematics Stochastics and Computation*" Cambridge University Press

Kijima, Masaaki (2013): "*Stochastic Processes with Application to Finance*" second addition. CRC Press, Taylor & Francis Group. 3 Park Square, Milton Park. Abingdon, Oxfordshire OX14 4RN, UK

Korn, Ralf; Korn, Elke (2001): *"Option Pricing and Portfolio Optimization, Modern Methods of Financial Mathematic".* Graduate Studies in Mathematics Volume 31. American Mathematical Society Providence, Rhode Island

Kyng, Timothy J.; Purcal, Sachi; and Zhang, Jinhui C. (2016): Excel implementation of finite difference methods for option pricing, *Spreadsheets in Education (eJSiE)*: Vol. 9: Iss. 3, Article 2.

Lin, J. & Liang, J. (2007) "Pricing of Perpetual American and Bermudan Options by Binomial Tree Method". *Front. Math. China* (2007) 2: 243.

Longstaff, Francis A.; Schwartz, Eduardo S. (2001): "Valuing American Options by Simulation: A Simple Least-Squares Approach", *The Review of Financial Studies Spring* 2001 Vol. No. 1, pp. 113-147

McDonald, Robert L (2013): *"Derivatives Markets: Pearson New International Edition".* Pearson Education Limited

Schwartz, Eduardo S. (1977): "The Valuation of Warrants: Implementing a New Approach". *Journal of Financial Economics 4* (1997) 79-93.

Stentoft, Lars (2004): "Assessing the Least Squares Monte-Carlo Approach to American Option Valuation". *Review of Derivatives Research, 7,* 129-168, 2004

White, H., (1984) *Asymptotic Theory for Econometricians,* Academic Press, New York

7 Appendix

7.1 STATA: Generating a Stock Price Path to Use in the LSM

Below is a program to generate stock price paths. To make your own example some parameters in the program can be changed. Observations is the length of the price path that one can change if a longer time frame is needed. The initial price s can be adjusted, as well as μ, σ^2 and δt. It is important to have the same value in *set obs 4* as in the command *forvalues i=2/4*. By running this program for example eight times, one obtain eight different price paths.

```
//program ppath1
drop _all /* clear workspace */
set obs 4 /* length of price path */
set matsize 1000 /* reserve space for matrix */
gen xi=rnormal(0,1) /* generate pseudo normal realizations */
gen s=1 /* intialize price path (initital price) */
//--------------------------input----------------------------------------------
scalar mu=0.05
scalar var=0.1024
scalar del=1/3
//-------------------------------------------------------------------------
scalar sqrtdel=sqrt(del) /* square root of delta */
scalar std=sqrt(var) /* compute standard deviation */
mkmat xi, matrix(e) /* convert variable e to matrix xi */
mkmat s, matrix(S) /* convert variable s to matrix S */
matrix list e
matrix list S
forvalues i=2/4 {
matrix S[‘i’,1] = S[‘i’-1,1]*exp((mu-0.5*var)*del+ std*sqrtdel*e[‘i’,1]) }
matrix list S
svmat S /* make matrix S into a STATA variable S1 */
list S1
gen t=_n
tsset t
tsline S1
//end
```

7.2 STATA: Program to Value an American Put Option in the Least Squares Monte Carlo Method

This program values American and European put options. Enter these commands in a do-file in STATA and press do. Remember to create the stock price paths in the data editor before running this program.

```
//--------------------------LSAM-----------------------------------
// this do file performs all computations necessary to generate
// the example on pages 32-39.
// ---------------------- t = 3 -----------------------------------------
gen sp=1.05
gen P3=sp-t3
replace P3=0 if P3<0
gen d0=exp(-0.05*3)*P3
summarize d0            // the mean is the value of the EU option
// ----------------------- t = 2 -----------------------------------------
gen xt2 = t2
replace xt2=. if xt2>sp    // eliminate out-of-money cases
gen y2=exp(-0.05*1)*P3
gen xt22=xt2^2
regress y2 xt2 xt22        // estimate polynomial regression
predict y2hat, xb
gen ex2=sp-xt2
replace ex2=0 if ex2<y2hat
gen P2=ex2                // set up P2
replace P2=0 if ex2==.
replace P3=0 if P2>0      // update P3
// ----------------------- t = 1 -----------------------------------------
gen xt1 = t1
replace xt1=. if xt1>sp    // eliminate out-of-money cases
gen y1=exp(-0.05*1)*P2
gen xt12=xt1^2
regress y1 xt1 xt12        // estimate polynomial regression
predict y1hat, xb
gen ex1=sp-xt1
replace ex1=0 if ex1<y1hat
```

```
gen P1=ex1
replace P1=0 if ex1==.
replace P2=0 if P1>0      // update P2
replace P3=0 if P1>0      // update P3
//------------------------- t = 0 -----------------------------------------
gen  pd3 =  exp(-0.05*3 )*P3
gen  pd2 =  exp(-0.05*2 )*P2
gen  pd1 =  exp(-0.05*1 )*P1
gen total = pd1 + pd2 + pd3
summarize total          // the mean gives the value of the AM option
//---------------------------------------------------------------------
```

7.3 MATLAB: Program to value an American Put Option in the Binomial Method (Cox, Ross and Rubinstein)

A zip file containing all the programs used in Matlab can be found here Pricing American Options by Mark Hoyle (2016).

To be able to value the American put option using the Matlab programs, one should enter the function as shown here

```
function [Price,P,S,Time] =
AmericanOptCRR();S0=100;K=105;r=0.05;T=1;sigma=0.2;N=50;type=true;
```

and in line 16, N needs to have the same value as in the function to run this program. This applies for Matlab programs in Appendix (7.4) and (7.5).

```
function [Price,P,S,Time] = AmericanOptCRR(S0,K,r,T,sigma,N,type)
%AmericanOptCRR - Price an American option via Cox-Ross-Rubinstein tree
%
% Returns the price of an American option computed using finite
% difference method applied to the Black Scholes PDE.
%
% Inputs:
%
% S0 Initial asset price
% K Strike Price
% r Interest rate
% T Time to maturity of option
% sigma Volatility of underlying asset
% N Number of points in time grid to use (minimum is 2, default is 50)
% type True (default) for a put, false for a call
if nargin < 6 || isempty(N), N = 50; end
if nargin < 7, type = true; end
dt = T/N;
u = exp(sigma*sqrt(dt)); d = 1/u;
a = exp(r*dt); p = (a-d)/(u-d);
% Create final Returns on the tree S{N+1} = S0*u^N*d.^(0:2:2*N);
if type
% Put option
P{N+1} = max(K-S{N+1},0);
```

```
else
P{N+1} = max(S{N+1}-K,0);
end
Time{N+1} = T*ones(1,N+1);
% Now move back through time and calculate the expected return at previous
% nodes on the tree. Compare this with the immediate return. Exercise the
% option if the immediate return is greater than the expected return
for ii = N:-1:1
Q = zeros(1,ii);
V = zeros(1,ii);
for jj = 1:ii
% Share price at current node
V(jj) = S0*u^(ii-1)*d^(2*(jj-1));
% Expected value of option due if we continue to hold
E = p*P{ii+1}(jj)/a+(1-p)*P{ii+1}(jj+1)/a;
% Value of early exercise
if type
% Put option
I = max(K-V(jj),0);
else
I = max(V(jj-K),0);
end
% Value of option at this Node
Q(jj) = max(E,I);
end
S{ii} = V;
P{ii} = Q;
Time{ii} = ii*dt*ones(size(S{ii}));
end
Price = P{1};
P = [P{:}];
S = [S{:}];
Time = [Time{:}];
```

7.4 MATLAB: Program to Value an American Put Option in the Finite Difference Method

A zip file containing all the programs used in Matlab can be found here Pricing American Options by Mark Hoyle (2016).

```
function [P_FD,P,s,t] = AmericanOptFD(S0,K,r,T,sigma,N,M,type)
%AmericanOptFD - Price an American option via finite differences
%
% Returns the price of an American option computed using finite
% difference method applied to the Black Scholes PDE.
%
% Inputs:
%
% S0 Initial asset price
% K Strike Price
% r Interest rate
% T Time to maturity of option
% sigma Volatility of underlying asset
% N Number of points in time grid to use (minimum is 3, default is 50)
% M Number of points in asset price grid to use (minimum is 3, default is 50)
% type True (default) for a put, false for a call
if nargin < 6 || isempty(N), N = 50; elseif N < 3, error('N has to be at least 3'); end
if nargin < 7 || isempty(M), M = 50; elseif M < 3, error('M has to be at least 3'); end
if nargin < 8, type = true; end
% create time grid
t = linspace(0,T,N+1);
dt = T/N; % Time step
% Share price grid
Smax = 2*max(S0,K)*exp(r*T); % Maximum price considered
dS = Smax/(M);
s = 0:dS:Smax;
% Now find points either side of the initial price so that we can calculate
% the price of the option via interpolation
idx = find(s < S0); idx = idx(end); a = S0-s(idx); b = s(idx+1)-S0;
Z = 1/(a+b)*[a b]; % Interpolation vector
% Set up a pricing matrix to hold the values we compute
```

```
P = NaN*ones(N+1,M+1); % Pricing Matrix (t,S)
% Boundary condition
if type
P(end,:) = max(K-(0:M)*dS,0); % Value of option at maturity - Put
else
P(end,:) = max((0:M)*dS-K,0); % Value of option at maturity - Call
end
P(:,1) = K; % Value of option when stock price is 0)
P(:,end) = 0; % Value of option when S = Smax
% Create matrix for finite difference calculations
J = (1:M-1)';
a = r/2*J*dt-1/2*sigma^2*J.^2*dt;
b = 1+sigma^2*J.^2*dt+r*dt;
c = -r/2*J*dt-1/2*sigma^2*J.^2*dt;
D = spdiags([[a(2:end);0] b [0;c(1:end-1)]],[-1 0 1],M-1,M-1);
% Finite difference solver
for ii = N:-1:1
y = P(ii+1,2:end-1)'+[-a(1)*K; zeros(M-3,1); -c(end)*0];
x = D\y; % Value of the option
if type
P(ii,2:end-1) = max(x,K-s(2:end-1)'); % Put
else P(ii,2:end-1) = max(x,s(2:end-1)'-K); % Call
end
end
% Extract the final price P_FD = Z*P(1,idx:idx+1)';
end
```

7.5 MATLAB: Program to value an American put Option in the Least Squares Monte Carlo Method

A zip file containing all the programs used in Matlab to price American options can be found here Pricing American Options by Mark Hoyle (2016).

```
function [Price,CF,S,t] = AmericanOptLSM(S0,K,r,T,sigma,N,M,type)
%AmericanOptLSM - Price an American option via Longstaff-Schwartz Method
%
% Returns the price of an American option computed using finite
% difference method applied to the Black Scholes PDE.
%
% Inputs:
%
% S0 Initial asset price
% K Strike Price
% r Interest rate
% T Time to maturity of option
% sigma Volatility of underlying asset
% N Number of points in time grid to use (minimum is 3, default is 50)
% M Number of points in asset price grid to use (minimum is 3, default is 50)
% type True (default) for a put, false for a call
if nargin < 6 || isempty(N), N = 50; elseif N < 3, error('N has to be at least 3'); end
if nargin < 7 || isempty(M), M = 50; elseif M < 3, error('M has to be at least 3'); end
if nargin < 8, type = true; end
dt = T/N;
t = 0:dt:T;
t = repmat(t',1,M);
R = exp((r-sigma^2/2)*dt+sigma*sqrt(dt)*randn(N,M));
S = cumprod([S0*ones(1,M); R]);
ExTime = (M+1)*ones(N,1);
% Now for the algorithm
CF = zeros(size(S)); % Cash flow matrix
CF(end,:) = max(K-S(end,:),0); % Option only pays off if it is in the money
for ii = size(S)-1:-1:2
if type
Idx = find(S(ii,:) < K); % Find paths that are in the money at time ii
```

```
else
Idx = find(S(ii,:) > K); % Find paths that are in the money at time ii
end
X = S(ii,Idx)'; X1 = X/S0;
Y = CF(ii+1,Idx)'*exp(-r*dt); % Discounted cashflow
R = [ ones(size(X1)) (1-X1) 1/2*(2-4*X1-X1.^2)];
a = R\Y; % Linear regression step
C = R*a; % Cash flows as predicted by the model
if type
Jdx = max(K-X,0) > C; % Immediate exercise better than predicted cashflow
else
Jdx = max(X-K,0) > C; % Immediate exercise better than predicted cashflow
end
nIdx = setdiff((1:M),Idx(Jdx));
CF(ii,Idx(Jdx)) = max(K-X(Jdx),0);
ExTime(Idx(Jdx)) = ii;
CF(ii,nIdx) = exp(-r*dt)*CF(ii+1,nIdx);
end
Price = mean(CF(2,:))*exp(-r*dt);
end
```

7.6 MATLAB: Program to Compare the Valuation Methods

A zip file containing all the programs used in Matlab can be found here Pricing American Options by Mark Hoyle (2016).

Below is the program to compare the valuation methods in terms of computational time. $S0$ is the initial stock price. K is the strike price. T is the time to maturity. $sigma$ is the volatility. M is the number of stock price steps (finite difference)/ length of price path (least squares Monte Carlo). N is the time steps. By changing these parameters one can obtain a graph of the difference in computational time and 3D plots for comparison of the valuation methods.

```
%% Compare the various methods
% Try up to 100 timesteps and compare the results in terms of time taken
% and how they agree
S0 = 90; K = 100; r = 0.05; T = 1; sigma = 0.2;
Timings = zeros(98,3);
Results = zeros(98,3);
M = 10000; % Number of grid spacings/MC paths for LSM and FD
for N = 3:100
tic;
Results(N-1,1) = AmericanOptCRR(S0,K,r,T,sigma,N);
Timings(N-1,1) = toc;
tic;
Results(N-1,2) = AmericanOptFD(S0,K,r,T,sigma,N,M);
Timings(N-1,2) = toc;
tic;
Results(N-1,3) = AmericanOptLSM(S0,K,r,T,sigma,N,M);
Timings(N-1,3) = toc;
end
%% Plot the results of this
subplot(2,1,1);
plot(Results);
grid
title('Option price','fontsize',14);
xlabel('Number of timesteps','fontsize',14);
ylabel('Option price','fontsize',14);
legend('CRR','FD','LSM','location','SE');
```

```
subplot(2,1,2);
plot(Timings);
grid
title('Time taken to compute','fontsize',14);
xlabel('Number of timesteps','fontsize',14);
ylabel('Method timings','fontsize',14);
legend('CRR','FD','LSM','location','NW');

%% How do they compare over a surface?
[Price,Pcrr,Scrr,Tcrr] = AmericanOptCRR(S0,K,r,T,sigma,100);
[Price,Pfd,Sfd,Tfd] = AmericanOptFD(S0,K,r,T,sigma,100,100);
[Price,Plsm,Slsm,Tlsm] = AmericanOptLSM(S0,K,r,T,sigma,100,100);
figure;
surf(Sfd,Tfd,Pfd); shading interp
line(Slsm,Tlsm,Plsm,'linestyle','none','marker','.','color','r');
line(Scrr,Tcrr,Pcrr,'linestyle','none','marker','.','color','b');
data.CRR.P = Pcrr;
data.CRR.S = Scrr;
data.CRR.T = Tcrr;
data.FD.P = Pfd;
data.FD.S = Sfd;
data.FD.T = Tfd;
data.LSM.P = Plsm;
data.LSM.S = Slsm;
data.LSM.T = Tlsm;
save AMERICAN_OPTION_DATA data;
```